

Universitat de Lleida
Escola Politècnica Superior
Enginyeria Tècnica en Informàtica de Sistemes

Treball de final de Carrera

**Disseny i implementació d'una aplicació web que
modelitza competicions esportives de Judo.**

Autor: Miquel Àngel Castellanos Larrégola
Director: Josep Maria Ribó i Balust
Setembre, 2007

ÍNDEX

1.	INTRODUCCIÓ	1
2.	APLICACIÓ INFOJUDO	2
2.1.	BREU INTRODUCCIÓ AL MÓN DEL JUDO	2
2.2.	EL JUDO A LLEIDA.....	3
2.3.	FUNCIONALITAT D'INFOJUDO	4
2.4.	TECNOLOGIES EMPRADES.....	4
3.	ANÀLISI DE REQUERIMENTS.....	5
3.1.	REQUERIMENTS GENERALS DEL SISTEMA	5
3.2.	USUARIS DE L' APLICACIÓ.....	6
3.2.1.	<i>Usuari de la federació.....</i>	<i>6</i>
3.2.2.	<i>Usuari de club</i>	<i>6</i>
3.2.3.	<i>Usuari genèric (usuari no registrat, usuari de la federació o usuari de club).....</i>	<i>6</i>
3.3.	DIAGRAMA DE CASOS D'ÚS	7
3.4.	CASOS D'ÚS	8
3.4.1.	<i>Crear una competició</i>	<i>8</i>
3.4.2.	<i>Afegir categories de pesos a una competició.....</i>	<i>9</i>
3.4.3.	<i>Esborrar una competició.....</i>	<i>10</i>
3.4.4.	<i>Modificar dades generals d'una competició.....</i>	<i>11</i>
3.4.5.	<i>Modificar resultats de competicions.....</i>	<i>12</i>
3.4.6.	<i>Calcular emparellaments d'una competició.....</i>	<i>13</i>
3.4.7.	<i>Afegir un club</i>	<i>14</i>
3.4.8.	<i>Esborrar un club.....</i>	<i>15</i>
3.4.9.	<i>Modificar les dades d'un club</i>	<i>16</i>
3.4.10.	<i>Donar d'alta un Judoka.....</i>	<i>17</i>
3.4.11.	<i>Modificació de les dades d'un judoka.....</i>	<i>18</i>
3.4.12.	<i>Inscriure un Judoka a una competició.....</i>	<i>19</i>
3.4.13.	<i>Esborrar un Judoka d'una competició.....</i>	<i>20</i>
3.4.14.	<i>Consultes.....</i>	<i>21</i>
3.5.	MODEL DE DOMINI	22
3.5.1.	<i>Explicació del model</i>	<i>23</i>
4.	ASPECTES DE DISSENY I IMPLEMENTACIÓ	24
4.1.	CLASSES DEL DISSENY	24
4.1.1.	<i>Classes de Model.....</i>	<i>24</i>
4.1.2.	<i>Classes de la Vista.....</i>	<i>26</i>
4.1.3.	<i>Classes del Controlador</i>	<i>27</i>
4.2.	DISSENY DE LA BASE DE DADES	29
5.	CONCLUSIONS I TREBALL FUTUR.....	31

A.	TECNOLOGIES WEB.....	31
A.1.	ARQUITECTURA MODEL/VISTA/CONTROLADOR	31
A.1.1.	<i>Definició de les parts.....</i>	32
A.2.	HTML	32
A.3.	JSP	34
A.4.	CSS.....	35
A.5.	JAVASCRIPT	36
B.	STRUTS.....	37
B.1.	STRUTS I MVC	37
B.2.	DIAGRAMA DE SEQÜÈNCIA.....	38
B.3.	COMPONENTS DE LA CAPA CONTROLADORA.....	38
C.	ESPECIFICACIÓ.....	39
C.1.	OPERACIONS DE LA CLASSE DBCOMPETITION.....	39
C.2.	OPERACIONS DE LA CLASSE DBJUDOKA.....	43
C.3.	OPERACIONS DE LA CLASSE DBCLUB.....	45
C.4.	OPERACIONS DE LA CLASSE DBRESULTCOMBATS.....	46
C.5.	OPERACIONS DE LA CLASSE DBCOMBATS.....	47
C.6.	OPERACIONS DE LA CLASSE DBPARTICIPANTS.....	48
C.7.	OPERACIONS DE LA CLASSE DBWEIGHTSCATEGORIES	48
C.8.	OPERACIONS DE LA CLASSE DBAGECATEGORIES	49
C.9.	OPERACIONS DE LA CLASSE DBWINNERS	49
D.	LLICÈNCIA GPL.....	50
	LLISTAT D'ACRÒNIMS	63
	BIBLIOGRAFIA	63

ÍNDEX DE FIGURES

IMATGE 1.	DIAGRAMA DE CASOS D'ÚS.....	7
IMATGE 2.	MODEL DE DOMINI.....	22
IMATGE 3.	ESQUEMA MODEL-VISTA-CONTROLADOR.....	32
IMATGE 4.	FUNCIONAMENT GENERAL TECNOLOGIA JSP.....	35

1. INTRODUCCIÓ

En el seu sentit més ampli, l'esport és qualsevol joc o activitat de lleure, basada en un reglament estricte i que procura millorar el benestar físic i mental de l'ésser humà.

El judo és un esport molt estès a Espanya i arreu del món, especialment en països com per exemple Japó, país on va néixer. Encara, però, un sector important de la població desconfia de la pràctica del judo ja que per desconeixença l'identifica com una simple art marcial.

El meu primer contacte amb el judo va ser als 8 anys. En aquella època era un jove soci del recentment obert Gimnàs Lleida, ara amb el nom d'Ekke, sens dubte un dels millors gimnasos de tota Espanya. Des d'aquesta tendra edat primer Pep Castarlenas i més endavant Go Tsunoda hem van inculcar els valors que un esport com el judo representa. La pràctica del judo i de la seva filosofia i la influència d'aquests dos grans entrenadors van ajudar a créixer i madurar a aquell nen i convertir-lo en la persona que és ara.

Fa uns anys, era un noi que somiava amb estudiar informàtica i un practicant assidu del judo. L'experiència que hem proporcionava assistir a diverses competicions i veure que la seva feina organitzativa es portava a terme a mà, unit a la gran afició mostrada per la informàtica, van fer que naixés per primera vegada la idea d'una aplicació relacionada amb el judo. Aquest projecte va ser l'excusa perfecta per a que aquesta idea agafés forma i es convertís en l'anomenada aplicació InfoJudo, que pretén facilitar l'organització de competicions per part de la Federació lleidatana de judo, i la difusió d'aquests esdeveniments per a que qualsevol persona des de casa seva els pugui consultar i els pugui conèixer millor. No es tracta d'un projecte acordat amb la Federació lleidatana, sinó que es tracta d'una motivació personal. No obstant, responsables de la Federació com Ra Badia o el propi president, Pep Eroles, han mostrat el seu interès en l'aplicació inclús demanant futures proves per testear-la.

Durant les pròximes pàgines es mostraran les diferents funcionalitats que posseeix l'aplicació InfoJudo, així com les tècniques utilitzades per la seva implementació.

2. APLICACIÓ INFOJUDO

El judo és actualment una de les arts marcials més esteses i respectades arreu del món. El fet que sigui olímpic sens dubte afavoreix que la cantera d'aquest esport augmenti cada any. Lleida no és una excepció en aquest marc. No obstant, com moltes altres vegades sembla que els recursos i tecnologies no creixent al mateix ritme que la demanda popular. InfoJudo neix amb la intenció de cobrir aquest espai.

2.1. Breu introducció al món del judo

El **judo** (柔道) és una art marcial japonesa fundada per Jigoro Kano a la fi del segle XIX a partir del Jujutsu, fugint de tècniques perilloses o inútils i sistematitzant-lo, essent així possible la seva pràctica lliure en *randori*. L'objectiu per Kano era aconseguir la millora de les persones. El *judo* ha aconseguit ser un esport olímpic (des dels Jocs Olímpics de Tokyo).

Per als més joves, que representa el 70 per cent de tots els practicants, la iniciació al judo, de caràcter lúdic, és una forma de descobrir el seu cos i de preparació a una veritable pràctica esportiva. D'adolescents, la seva pràctica serà més tècnica i de descobriment dels aspectes culturals... D'adult, el judoka podrà seguir la pràctica tradicional, espècie d'esport relax, d'equilibri físic. En la tercera edat, amb la condició d'establir algunes precaucions elementals, podran seguir l'estudi del judo al seu ritme, substituint cada vegada més la força muscular per l'harmonia en el gest.

El judo és un esport en el qual es veu involucrat tot el cos i la intensitat del qual i demanda cardiovascular poden ser modulats perfectament per l'esportista. Combina perfectament la força, la tàctica de combat i la tècnica, amb el que es produeix un desenvolupament integral de la persona. En ell, cal combinar una bona preparació física tant de tipus anaeròbic com aeròbic ja que a les accions explosives i de gran velocitat cal afegir una gran resistència per a aguantar la durada d'un combat.

Les tècniques del judo bàsicament són projeccions, immobilitzacions, luxacions i estrangulacions.

El judo actual ha perdut bona part de la cavallerositat originària i s'ha convertit en un esport de competició, en una baralla molt travada, en la qual és molt difícil veure la puresa de la tècnica que antigament s'ensenyava.

Els judoques, duen una vestimenta concreta, és el keikogi que en el Judo pren el nom de judogi, una jaqueta i un pantaló fets de tela resistent i generalment blanca i un cinturó (obi), tot plegat popularment conegut a occident com a kimono encara que no és pas el mateix.

Capítol 2: Aplicació InfoJudo

Els cinturons (obi) comencen pel color blanc i li segueixen el groc, taronja, verd, blau, marró i negre, passant també per uns colors intermedis, indiquen la progressió en el coneixement. Després del negre apareixen altres graduacions denominades “Dan”.

El lloc on es practica rep el nom de dojo i en el terra hi ha el tatami, uns matalassets per no fer-se mal al caure.

Les puntuacions existents en un combat són:

- Ippon: Consisteix en una projecció perfecta que acabaria amb l'esquena del contrincant contra el terra.
- Wazari: Es tracta d'una projecció de menor nivell que l'Ippon. L'esquena del contrincant també acabaria contra el terra però la tècnica utilitzada seria menys depurada que en un Ippon.
- Yuko: És una tècnica que portaria un dels dos costats de l'esquena de l'oponent contra el terra. L'esquena, per tant, no té un contacte total amb el terra.
- Koka: És la projecció de menor nivell de totes. L'oponent cau al terra en primer lloc amb el cul.

Tenint en compte aquestes puntuacions, un combat de Judo es pot arribar a guanyar de dues maneres diferents:

- Quan un dels dos participants realitza un Ippon. En algunes competicions es pot pujar el nombre d'Ippons necessaris per guanyar un combat.
- Per termini del temps. Habitualment els combats duren entre 2 i 4 minuts depenent de l'edat dels participants. Quan el temps d'un combat arriba a la fi i cap judoka ha aconseguit un Ippon, el guanyador serà aquell que tingui més nombre de Wazaris, Yukos o Kokes respectivament. En cas d'empat en les puntuacions seran els àrbitres qui decidiran al guanyador segons factors com la tècnica mostrada o el nivell d'intensitat en el combat.

2.2. *El judo a Lleida*

La tradició judoka a Lleida és extensa. L'acceptació social que aquesta art marcial té, va permetre que aviat molts gimnasos de la capital catalana oferissin classes de Judo. Aviat es va fer necessària l'organització de competicions i esdeveniments relacionats amb el judo.

Ara per ara la cantera de Judo a Lleida és gran i prometedora, no obstant, molts d'aquests judokes es perden pel camí i no arriben a les categories superiors. Això fa que les categories a partir dels 15 anys es trobin habitualment poc poblades. Per suplir aquesta carència les competicions es solen disputar en format lliga.

En una competició s'han de tenir en compte dos factors: l'edat i el pes dels participants. L'edat ja es té en compte en el moment que es declara que una competició és d'una categoria d'edat determinada, mentre que el pes s'haurà de tenir en compte a l'hora de distribuir als participants en les lligues.

Tota aquesta feina administrativa recau com és evident en la Federació. En l'actualitat aquestes feines organitzatives es porten a terme a mà amb la conseqüent ineficiència.

2.3. Funcionalitat d'InfoJudo

L'aplicació que es vol portar a terme, anomenada “InfoJudo” , busca informatitzar l'organització dels esdeveniments esportius per part de la Federació lleidatana de Judo, apropant al mateix temps les competicions de Judo al gran públic.

No es tracta d'un projecte oficial amb la Federació lleidatana de Judo encara que varis membres, entre ells el propi president, han mostrat el seu interès en l'aplicació.

2.4. Tecnologies emprades

L'aplicació web es troba implementada sobre la tecnologia de Struts. Per tant, s'ha utilitzat l'arquitectura MVC (Model Vista Controlador) que facilita la feina i afavoreix a possibles reciclatges de codi. La capa del Model utilitza llenguatge Java mentre que la Vista per mostrar les pàgines està implementada amb JSP (en format HTML) i maquetades amb CSS. També de cara a la capa de presentació s'ha utilitzat el llenguatge interpretat JavaScript.

S'aprofundirà més en totes aquestes tecnologies en els annexos (*Vegeu annex A*).

3. ANÀLISI DE REQUERIMENTS

L'anàlisi de requeriments és el primer pas en el disseny d'una aplicació. És l'etapa on s'aconsegueix claredat sobre el que desitja l'usuari i la forma en que es presentarà la solució que està buscant.

A partir de la definició dels requeriments generals i dels usuaris del sistema es podran redactar els casos d'us i desenvolupar un diagrama de model de domini que servirà com a base indispensable per a la implementació.

3.1. *Requeriments generals del sistema*

Es desitja que l'aplicació sigui capaç de modelitzar i administrar les competicions esportives de judo per a la Federació lleidatana d'aquest esport.

Per aquest motiu es vol que el sistema pugui crear competicions en format de lliga tenint en compte el pes dels participants.

Aquestes competicions poden ser masculines o femenines i a demés existiran 5 categories d'ambdues: sub-15, sub-17, sub-20, sub-23 i més de 23.

Les competicions poden tenir diverses categories de pesos. Els organitzadors de les competicions seran els que decidiran quines categories té cada competició encara que aquestes categories es trobaran limitades: -44 Kg, -48 Kg, -52 Kg, -55 Kg, -57 Kg, -60 Kg, -63 Kg, -66 Kg, -70 Kg, -73 Kg, -78 Kg, -81 Kg, -90 Kg, -100 Kg i més de 100 Kg.

De totes les competicions es volen tenir tots els seus combats (aparellaments) així com els seus resultats i algunes dades d'interès succeïdes en el seu transcurs: Tècnica emprada per guanyar, minut de la victòria i comentaris.

Es vol que aquestos aparellaments els calculi el sistema de forma automàtica tenint en compte els pesos dels participants.

La puntuació d'un combat en ordre descendent i la seva equivalència en punts és aquesta: Ippon (10 punts), Wazari (8 punts), Yuko (4 punts) i Koka (2 punts).

Si un club que no forma part del sistema ho vol fer haurà de demanar a la federació que l'inscrigui.

Quan un club vol inscriure un judoka a una competició l'haurà d'afegir prèviament al sistema i una vegada afegit només haurà de sol·licitar a través de l'aplicació la participació d'aquest judoka en tantes competicions com es desitgi. La federació pot esborrar d'una competició a un judoka que no compleixi amb els requisits establerts. Així mateix els clubs també podran retirar la inscripció d'un judoka del seu club a una competició.

Tothom qui vulgui podrà accedir a l'aplicació i consultar dades referents a les competicions, clubs o judokes.

Els treballadors de la Federació i dels clubs inscrits seran els únics que podran consultar dades personals sobre judokes.

3.2. Usuaris De l'aplicació

Existiran 3 tipus d'usuaris diferents: treballadors de la federació (administradors), treballadors de clubs i usuari no registrat.

3.2.1. Usuari de la federació

Serán els usuaris que tindran més control sobre l'aplicació actuant com a administradors del sistema.

Podran:

- Afegir/esborrar/modificar competicions.
- Afegir/esborrar/modificar dades d'un club.
- Modificar dades sobre judokes.
- Esborrar participants d'una competició.

3.2.2. Usuari de club

Aquest tipus d'usuari té més limitats els privilegis sobre l'aplicació. Es trobaran en condicions de:

- Afegir/esborrar judokes al sistema com a esportistes del seu club.
- Inscriure/esborrar judokes del seu club, afegits prèviament al sistema, a una competició existent.

3.2.3. Usuari genèric (usuari no registrat, usuari de la federació o usuari de club)

Els usuaris genèrics engloben tres rols diferents. Poden ser aquells usuaris que no s'han registrat al sistema i només desitgen realitzar una consulta o un dels dos usuaris registrats. Tots tres podran realitzar consultes. Les consultes que podran fer són:

- Consultar dades sobre competicions.

Els camps que tots els usuaris podran consultar de les competicions són: Nom, sexe, categoria d'edat i any de celebració.

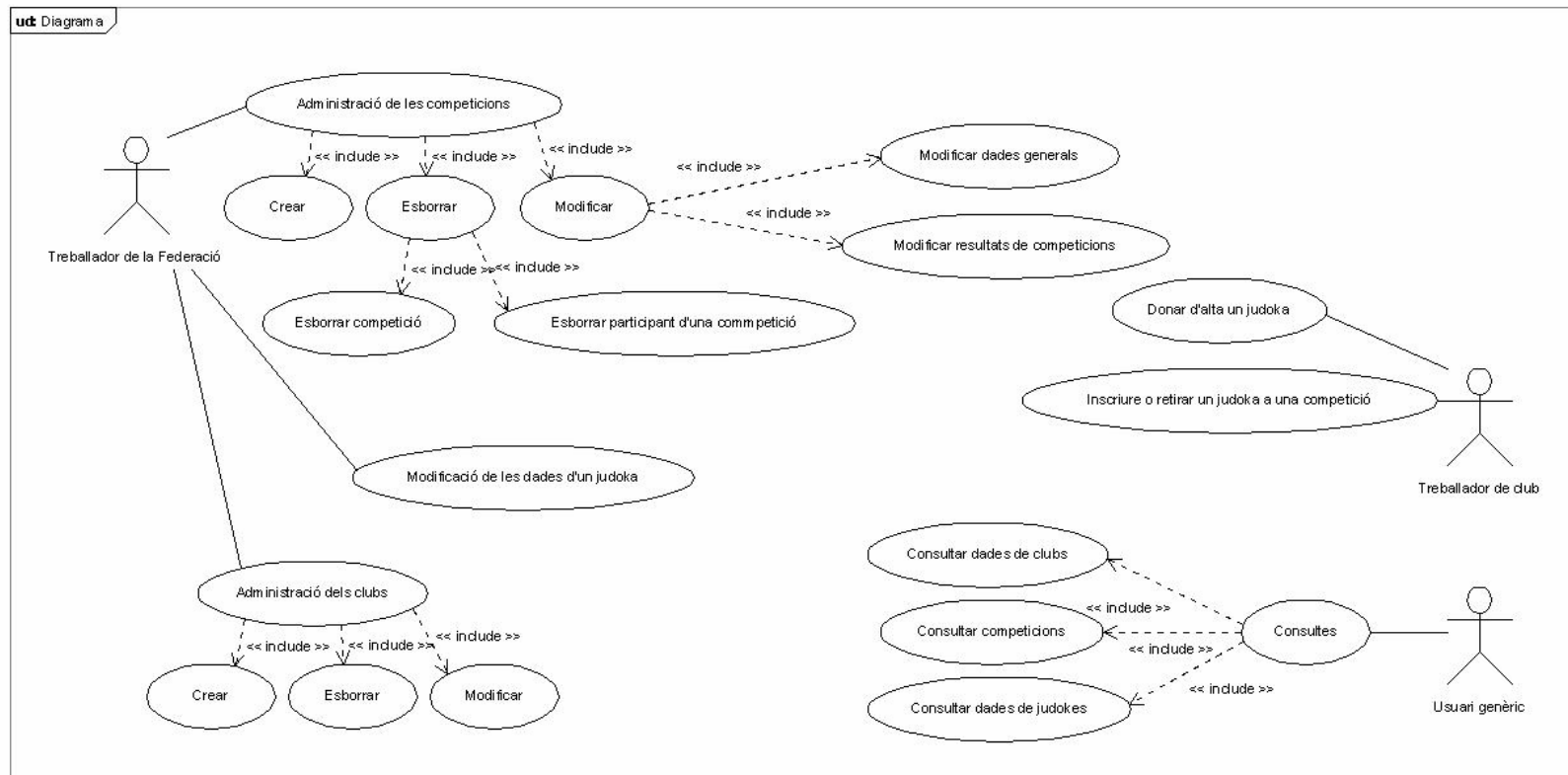
D'igual manera tots els usuaris podran consultar les següents dades sobre el lloc de celebració de les competicions: la seva direcció, el seu aforament, la seva localitat, la província, el país i comentaris.

- Consultar dades de judokes.
 - a) Si es tracta d'un usuari no registrat: Nom, primer cognom, segon cognom, cinturó, club del que és soci, data de naixement, sexe i comentaris.
 - b) Si es tracta d'un usuari registrat: A més de tot el que pot veure un usuari no registrat també podrà veure la direcció del judoka, el seu país, localitat, província, telèfons de contacte i email.
- Consultar dades de clubs.

Les dades que podran visualitzar tots els usuaris serán les següents: Nom del club, nombre de membres, any de fundació, fax, web, entrenador de judo, direcció, província, localitat, país i telèfons de contacte.

3.3. Diagrama De Casos d'ús

En el següent diagrama de casos d'ús es poden observar als usuaris abans esmentats i les operacions que poden realitzar.



Imatge 1. Diagrama de casos d'ús

3.4. Casos d'ús

Un cas d'ús especifica una seqüència d'accions, incloent variants, que el sistema pot executar i que produeix un resultat observable de valor per a un particular actor.

El seu component més important és reunir en tots els casos d'ús els requeriments del sistema de software que s'està construint.

Un cas d'ús descriu *què* fa el sistema, no *com* ho fa.

A continuació es descriuen els casos d'ús definits a partir dels requeriments del sistema descrits anteriorment.

3.4.1. Crear una competició

Cas d'ús: Crear una competició.

Objectius: Crear competicions.

Actors:

- Iniciador: treballador de la Federació

Tipus: Primari.

Precondicions: El treballador de la Federació s'ha autenticat correctament.

Postcondicions: La nova competició ha estat creada amb èxit.

Flux principal d'esdeveniments:

1. El treballador de la Federació s'autentica per entrar al sistema.
2. El sistema mostra el menú.
3. El treballador selecciona l'opció d'afegir competició.
4. El sistema mostra el formulari corresponent.
5. El treballador completa el formulari i l'envia.
6. El sistema crea i emmagatzema la nova competició.

Flux alternatiu d'esdeveniments:

1. El treballador de la Federació s'autentica per entrar al sistema.
 - 1.1. El sistema falla (en qualsevol moment).
 - 1.1.1. El sistema es reinicia demanant novament l'autenticació.
 - 1.2. L'autenticació és incorrecta.
 - 1.2.1. El sistema demana novament l'autenticació.
2. El sistema mostra el menú.
3. El treballador selecciona l'opció d'afegir competició.
4. El sistema mostra el formulari corresponent.
5. El treballador completa el formulari i l'envia.
6. El sistema crea i emmagatzema la nova competició.
 - 6.1. Apareix un error al emmagatzemar les dades.
 - 6.1.1. Es torna a mostrar el formulari de la creació de competicions i es demana que es tornin a introduir les dades.

3.4.2. Afegir categories de pesos a una competició

Cas d'ús: Afegir categories de pesos a una competició.

Objectius: Afegir les categories de pesos a una competició que s'ha creat amb anterioritat.

Actors:

- Iniciador: treballador de la Federació

Tipus: Primari.

Precondicions: El treballador de la Federació s'ha autenticat correctament i la competició a la que es vol afegir les categories ja existeix.

Postcondicions: La competició té assignades les categories de pesos que s'han seleccionat.

Flux principal d'esdeveniments:

1. El treballador de la Federació s'autentica per entrar al sistema.
2. El sistema mostra el menú.
3. El treballador busca la competició que desitja mitjançant el seu nom o any de celebració i prem l'opció d'editar.
4. El sistema mostra les dades de la competició.
5. El treballador accedeix a l'apartat de categoria de pesos, selecciona una categoria i prem el botó d'afegir.
6. El sistema relaciona la categoria amb la competició.

Flux alternatiu d'esdeveniments:

1. El treballador de la Federació s'autentica per entrar al sistema.
 - 1.1. El sistema falla (en qualsevol moment).
 - 1.1.1. El sistema es reinicia demanant novament l'autenticació.
 - 1.2. L'autenticació és incorrecta.
 - 1.2.1. El sistema demana novament l'autenticació.
2. El sistema mostra el menú.
3. El treballador busca la competició que desitja mitjançant el seu nom o any de celebració i prem l'opció d'editar.
 - 3.1. La competició que busca el treballador no existeix.
 - 3.1.1. El sistema retorna un missatge d'error explicant la situació.
4. El sistema mostra les dades de la competició.
5. El treballador accedeix a l'apartat de categoria de pesos, selecciona una categoria i prem el botó d'afegir.
 - 5.1. La categoria de pesos ja s'havia relacionat amb la competició amb anterioritat.
 - 5.1.1. El sistema retorna un missatge explicant la situació.
6. El sistema relaciona la categoria amb la competició.
 - 6.1. Apareix un error al relacionar.
 - 6.1.1. S'escriu un missatge explicant la situació.

3.4.3. Esborrar una competició

Cas d'ús: Esborrar una competició.

Objectius: Esborrar competicions.

Actors:

- Iniciador: treballador de la Federació

Tipus: Primari.

Precondicions: El treballador de la Federació s'ha autenticat correctament.

Postcondicions: La nova competició ha estat esborrada amb èxit.

Flux principal d'esdeveniments:

1. El treballador de la Federació s'autentica per entrar al sistema.
2. El sistema mostra el menú.
3. El treballador accedeix a l'apartat dedicat a les competicions.
4. El treballador busca la competició que desitja mitjançant el seu nom o any de celebració i li demana al sistema que l'esborri.
5. El sistema esborra la competició.

Flux alternatiu d'esdeveniments:

1. El treballador de la Federació s'autentica per entrar al sistema.
 - 1.1. El sistema falla (en qualsevol moment).
 - 1.1.1. El sistema es reinicia demanant novament l'autenticació.
 - 1.2. L'autenticació és incorrecta.
 - 1.2.1. El sistema demana novament l'autenticació.
2. El sistema mostra el menú.
3. El treballador accedeix a l'apartat dedicat a les competicions.
4. El treballador busca la competició que desitja mitjançant el seu nom o any de celebració i li demana al sistema que l'esborri.
 - 4.1. La competició que busca el treballador no existeix.
 - 4.1.1. El sistema retorna un missatge d'error explicant la situació.
5. El sistema esborra la competició.
 - 5.1. Apareix un error al esborrar la competició.
 - 5.1.1. El sistema retorna un missatge d'error explicant la situació.

3.4.4. Modificar dades generals d'una competició

Cas d'ús: Modificar dades generals d'una competició.

Objectius: Modificar les dades tals com la localitat on es celebren, el tipus de competició que és, etc.

Actors:

- Iniciador: treballador de la Federació

Tipus: Primari.

Precondicions: El treballador de la Federació s'ha autenticat correctament.

Postcondicions: Les noves dades de la competició s'han emmagatzemat amb èxit.

Flux principal d'esdeveniments:

1. El treballador de la Federació s'autentica per entrar al sistema.
2. El sistema mostra el menú.
3. El treballador accedeix a l'apartat dedicat a les competicions.
4. El treballador busca la competició que desitja mitjançant el seu nom o any de celebració i selecciona l'opció d'editar.
5. El sistema mostra les dades.
6. El treballador modifica les dades desitjades i les envia al sistema.
7. El sistema emmagatzema les noves dades.

Flux alternatiu d'esdeveniments:

1. El treballador de la Federació s'autentica per entrar al sistema.
 - 1.1. El sistema falla (en qualsevol moment).
 - 1.1.1. El sistema es reinicia demanant novament l'autenticació.
 - 1.2. L'autenticació és incorrecta.
 - 1.2.1. El sistema demana novament l'autenticació.
2. El sistema mostra el menú.
3. El treballador accedeix a l'apartat dedicat a les competicions.
4. El treballador busca la competició que desitja mitjançant el seu nom o any de celebració i selecciona l'opció d'editar.
 - 4.1. La competició que busca el treballador no existeix.
 - 4.1.1. El sistema retorna un missatge d'error explicant la situació.
5. El sistema mostra les dades.
6. El treballador modifica les dades desitjades i les envia al sistema.
7. El sistema emmagatzema les noves dades.
 - 7.1. Apareix un error al guardar les noves dades.
 - 7.1.1. El sistema retorna un missatge d'error explicant la situació.

3.4.5. Modificar resultats de competicions

Cas d'ús: Modificar resultats de competicions.

Objectius: Modificar els resultats dels combats d'una competició.

Actors:

- Iniciador: treballador de la Federació

Tipus: Primari.

Precondicions: El treballador de la Federació s'ha autenticat correctament.

Postcondicions: Els nous resultats dels combats s'han emmagatzemat amb èxit.

Flux principal d'esdeveniments:

1. El treballador de la Federació s'autentica per entrar al sistema.
2. El sistema mostra el menú.
3. El treballador accedeix a l'apartat dedicat a les competicions.
4. El treballador busca la competició que desitja mitjançant el seu nom o any de celebració i accedeix als seus combats.
5. Modifica els resultats dels combats desitjats i els envia al sistema.
6. El sistema emmagatzema els nous resultats.

Flux alternatiu d'esdeveniments:

1. El treballador de la Federació s'autentica per entrar al sistema.
 - 1.1. El sistema falla (en qualsevol moment).
 - 1.1.1. El sistema es reinicia demanant novament l'autenticació.
 - 1.2. L'autenticació és incorrecta.
 - 1.2.1. El sistema demana novament l'autenticació.
2. El sistema mostra el menú.
3. El treballador accedeix a l'apartat dedicat a les competicions.
4. El treballador busca la competició que desitja mitjançant el seu nom o any de celebració i accedeix als seus combats.
 - 4.1. La competició que busca el treballador no existeix.
 - 4.1.1. El sistema retorna un missatge d'error explicant la situació.
5. Modifica els resultats dels combats desitjats i els envia al sistema.
6. El sistema emmagatzema els nous resultats.
 - 6.1. Apareix un error al guardar les noves dades.
 - 6.1.1. El sistema retorna un missatge d'error explicant la situació.

3.4.6. Calcular emparellaments d'una competició

Cas d'ús: Calcular emparellaments d'una competició.

Objectius: Crear els emparellaments d'una competició tenint en compte el tipus de competició que és i el pes dels seus participants.

Actors:

- Iniciador: treballador de la Federació

Tipus: Primari.

Precondicions: Tots els pesos dels participants han estat introduïts així com el tipus de competició. El treballador de la Federació s'autentica correctament per accedir al sistema.

Postcondicions: La competició té creats tots els seus emparellaments.

Flux principal d'esdeveniments:

1. El treballador de la Federació s'autentica i accedeix al sistema.
2. El sistema mostra el menú.
3. El treballador accedeix a l'apartat dedicat a les competicions.
4. El treballador busca la competició que desitja mitjançant el seu nom o any de celebració i accedeix als seus combats.
5. El treballador demana que el sistema calculi els emparellaments.
6. El sistema calcula els emparellaments i els emmagatzema.

Flux alternatiu d'esdeveniments:

1. El treballador de la Federació s'autentica i accedeix al sistema.
 - 1.1. El sistema falla (en qualsevol moment).
 - 1.1.1. El sistema es reinicia demanant novament l'autenticació.
 - 1.2. L'autenticació és incorrecta.
 - 1.2.1. El sistema demana novament l'autenticació.
2. El sistema mostra el menú.
3. El treballador accedeix a l'apartat dedicat a les competicions.
4. El treballador busca la competició que desitja mitjançant el seu nom o any de celebració i accedeix als seus combats.
 - 4.1. La competició que busca el treballador no existeix.
 - 4.1.1. El sistema retorna un missatge d'error explicant la situació.
5. El treballador demana que el sistema calculi els emparellaments.
6. El sistema calcula els emparellaments i els emmagatzema.
 - 6.1. Apareix un error al guardar les noves dades.
 - 6.1.1. El sistema retorna un missatge d'error explicant la situació.

3.4.7. Afegir un club

Cas d'ús: Afegir un club.

Objectius: Afegir un nou club al sistema.

Actors:

- Iniciador: treballador de la Federació

Tipus: Primari.

Precondicions: Per afegir un club el treballador de la Federació necessita que li hagi demanat el propi club. El treballador s'autentica correctament.

Postcondicions: El club ha estat afegit al sistema com un nou club.

Flux principal d'esdeveniments:

1. El treballador de la Federació s'autentica i accedeix al sistema.
2. El sistema mostra el menú.
3. El treballador selecciona l'opció d'afegir club.
4. El sistema mostra el formulari corresponent.
5. El treballador completa el formulari i l'envia.
6. El sistema crea i emmagatzema el nou club.

Flux alternatiu d'esdeveniments:

1. El treballador de la Federació s'autentica i accedeix al sistema.
 - 1.1. El sistema falla (en qualsevol moment).
 - 1.1.1. El sistema es reinicia demanant novament l'autenticació.
 - 1.2. L'autenticació és incorrecta.
 - 1.2.1. El sistema demana novament l'autenticació.
2. El sistema mostra el menú.
3. El treballador accedeix a l'apartat dedicat als clubs.
4. El treballador demana el formulari de creació de clubs.
5. Introdueix les dades i les envia al sistema.
6. El sistema emmagatzema el nou club.
 - 6.1. Apareix un error al guardar les noves dades.
 - 6.1.1. El sistema retorna un missatge d'error explicant la situació.

3.4.8. Esborrar un club

Cas d'ús: Esborrar un club.

Objectius: Esborrar un club del sistema.

Actors:

- Iniciador: treballador de la Federació

Tipus: Primari.

Precondicions: Per esborrar un club el treballador de la Federació necessita que li hagi demanat el propi club. El treballador s'autentica correctament.

Postcondicions: El club ha estat esborrat del sistema.

Flux principal d'esdeveniments:

1. El treballador de la Federació s'autentica i accedeix al sistema.
2. El sistema mostra el menú.
3. El treballador accedeix a l'apartat dedicat als clubs.
4. El treballador busca el club que desitja mitjançant el seu nom i li demana al sistema que l'esborri.
5. El sistema esborra el club.

Flux alternatiu d'esdeveniments:

1. El treballador de la Federació s'autentica per entrar al sistema.
 - 1.1. El sistema falla (en qualsevol moment).
 - 1.1.1. El sistema es reinicia demanant novament l'autenticació.
 - 1.2. L'autenticació és incorrecta.
 - 1.2.1. El sistema demana novament l'autenticació.
2. El sistema mostra el menú.
3. El treballador accedeix a l'apartat dedicat als clubs.
4. El treballador busca el club que desitja mitjançant el seu nom i li demana al sistema que l'esborri.
 - 4.1. El club que busca el treballador no existeix.
 - 4.1.1. El sistema retorna un missatge d'error explicant la situació.
5. El sistema esborra el club.
 - 5.1. Apareix un error al esborrar el club.
 - 5.1.1. El sistema retorna un missatge d'error explicant la situació.
 - 5.1.2. Es torna a seleccionar el club desitjat i es torna a demanar al sistema que l'esborri.

3.4.9. Modificar les dades d'un club

Cas d'ús: Modificar les dades d'un club.

Objectius: Modificar les dades d'un club com per exemple el seu telèfon, nombre de socis, email, etc.

Actors:

- Iniciador: treballador de la Federació

Tipus: Primari.

Precondicions: Per modificar les dades d'un club el treballador de la Federació necessita que li hagi demanat el propi club. El treballador s'autentica correctament.

Postcondicions: Les dades modificades del club han estat correctament emmagatzemades al sistema.

Flux principal d'esdeveniments:

1. El treballador de la Federació s'autentica i accedeix al sistema.
2. El sistema mostra el menú.
3. El treballador accedeix a l'apartat dedicat als clubs.
4. El treballador busca el club que desitja mitjançant el seu nom i selecciona l'opció d'editar.
5. El sistema mostra les dades.
6. El treballador modifica les dades desitjades i les envia al sistema.
7. El sistema emmagatzema les noves dades.

Flux alternatiu d'esdeveniments:

1. El treballador de la Federació s'autentica i accedeix al sistema.
 - 1.1. El sistema falla (en qualsevol moment).
 - 1.1.1. El sistema es reinicia demanant novament l'autenticació.
 - 1.2. L'autenticació és incorrecta.
 - 1.2.1. El sistema demana novament l'autenticació.
2. El sistema mostra el menú.
3. El treballador accedeix a l'apartat dedicat als clubs.
4. El treballador busca el club que desitja mitjançant el seu nom i selecciona l'opció d'editar.
 - 4.1. El club que busca el treballador no existeix.
 - 4.1.1. El sistema retorna un missatge d'error explicant la situació.
5. El sistema mostra les dades.
6. El treballador modifica les dades desitjades i les envia al sistema.
7. El sistema emmagatzema les noves dades.
 - 7.1. Apareix un error al guardar les noves dades.
 - 7.1.1. El sistema retorna un missatge d'error explicant la situació.

3.4.10. Donar d'alta un Judoka

Cas d'ús: Donar d'alta un Judoka.

Objectius: Afegir un nou Judoka al sistema.

Actors:

- Iniciador: treballador del club

Tipus: Primari.

Precondicions: El club que donarà d'alta al Judoka té el consentiment d'aquest Judoka per fer-ho. A més el treballador del club s'ha d'autenticar correctament al sistema.

Postcondicions: El judoka està afegit al sistema amb les seves dades com a judoka completades.

Flux principal d'esdeveniments:

1. El treballador de la Federació s'autentica i accedeix al sistema.
2. El sistema mostra el menú.
3. El treballador accedeix a l'apartat dedicat als judokes.
4. El treballador demana el formulari de creació de judokes.
5. Introdueix les dades i les envia al sistema.
6. El sistema emmagatzema al nou judoka.

Flux alternatiu d'esdeveniments:

1. El treballador de la Federació s'autentica i accedeix al sistema.
 - 1.1. El sistema falla (en qualsevol moment).
 - 1.1.1. El sistema es reinicia demanant novament l'autenticació.
 - 1.2. L'autenticació és incorrecta.
 - 1.2.1. El sistema demana novament l'autenticació.
2. El sistema mostra el menú.
3. El treballador accedeix a l'apartat dedicat als judokes.
4. El treballador demana el formulari de creació de judokes.
5. Introdueix les dades i les envia al sistema.
6. El sistema emmagatzema al nou judoka.
 - 6.1. Apareix un error al guardar les noves dades.
 - 6.1.1. El sistema retorna un missatge d'error explicant la situació.

3.4.11. Modificació de les dades d'un judoka

Cas d'ús: Modificació de les dades d'un judoka.

Objectius: Modificar dades dels judokes com cinturó, pes, nom, adreça, etc.

Actors:

- Treballador de la Federació o treballador de club

Tipus: Primari.

Precondicions: El treballador de la Federació s'autentica correctament.

Postcondicions: Les dades del judoka han estat modificades i correctament emmagatzemades.

Flux principal d'esdeveniments:

1. El treballador de la Federació s'autentica i accedeix al sistema.
2. El sistema mostra el menú.
3. El treballador accedeix a l'apartat dedicat als judokes.
4. El treballador busca el judoka que desitja mitjançant el seu cognom o DNI i selecciona l'opció d'editar
5. El sistema mostra les dades.
6. El treballador modifica les dades desitjades i les envia al sistema.
7. El sistema emmagatzema les noves dades.

Flux alternatiu d'esdeveniments:

1. El treballador de la Federació s'autentica i accedeix al sistema.
 - 1.1. El sistema falla (en qualsevol moment).
 - 1.1.1. El sistema es reinicia demanant novament l'autenticació.
 - 1.2. L'autenticació és incorrecta.
 - 1.2.1. El sistema demana novament l'autenticació.
2. El sistema mostra el menú.
3. El treballador accedeix a l'apartat dedicat als judokes.
4. El treballador busca el judoka que desitja mitjançant el seu cognom o DNI i selecciona l'opció d'editar.
 - 4.1. El judoka que busca el treballador no existeix.
 - 4.1.1. El sistema retorna un missatge d'error explicant la situació.
5. El sistema mostra les dades.
6. El treballador modifica les dades desitjades i les envia al sistema.
7. El sistema emmagatzema les noves dades.
 - 7.1. Apareix un error al guardar les noves dades.
 - 7.1.1. El sistema retorna un missatge d'error explicant la situació.

3.4.12. Inscriure un Judoka a una competició

Cas d'ús: Inscriure un Judoka a una competició.

Objectius: Inscriure un Judoka del propi club a una competició creada amb anterioritat per un treballador de la Federació.

Actors:

- Iniciador: treballador del club

Tipus: Primari.

Precondicions: El Judoka que es vol inscriure a la competició forma part del club que el vol inserir. El Judoka dóna el consentiment al seu club per ser inscrit. El treballador del club s'autentica correctament per accedir al sistema.

Postcondicions: La competició té com a nou participant el Judoka inscrit.

Flux principal d'esdeveniments:

1. El treballador de la Federació s'autentica i accedeix al sistema.
2. El sistema mostra el menú.
3. El treballador accedeix a l'apartat dedicat a les competicions.
4. El treballador busca la competició que desitja mitjançant el seu nom o any de celebració i demana al sistema el formulari amb les seves dades.
5. El treballador demana el llistat dels seus judokes i selecciona els que vulgui inscriure.
6. El treballador envia el formulari amb els judokes seleccionats.
7. El sistema emmagatzema les noves dades.

Flux alternatiu d'esdeveniments:

1. El treballador del club s'autentica i accedeix al sistema.
 - 1.1. El sistema falla (en qualsevol moment).
 - 1.1.1. El sistema es reinicia demanant novament l'autenticació.
 - 1.2. L'autenticació és incorrecta.
 - 1.2.1. El sistema demana novament l'autenticació.
2. El sistema mostra el menú.
3. El treballador accedeix a l'apartat dedicat a les competicions.
4. El treballador busca la competició que desitja mitjançant el seu nom o any de celebració i demana al sistema el formulari amb les seves dades.
 - 4.1. La competició que busca el treballador no existeix.
 - 4.1.1. El sistema retorna un missatge d'error explicant la situació.
5. El treballador demana el llistat dels seus judokes i selecciona els que vulgui inscriure.
6. El treballador envia el formulari amb els judokes seleccionats.
7. El sistema emmagatzema les noves dades.
 - 7.1. Apareix un error al guardar les noves dades.
 - 7.1.1. El sistema retorna un missatge d'error explicant la situació.

3.4.13. Esborrar un Judoka d'una competició

Cas d'ús: Esborrar un Judoka d'una competició.

Objectius: Esborrar un Judoka del propi club d'una competició creada amb anterioritat per un treballador de la Federació.

Actors:

- Iniciador: treballador del club

Tipus: Primari.

Precondicions: El Judoka que es vol esborrar de la competició forma part del club que el vol esborrar. El Judoka dóna el consentiment al seu club per ser esborrat. El treballador del club s'autentica correctament per accedir al sistema.

Postcondicions: La competició deixa de tenir com a participant al Judoka esborrat.

Flux principal d'esdeveniments:

1. El treballador del club s'autentica i accedeix al sistema.
2. El sistema mostra el menú.
3. El treballador accedeix a l'apartat dedicat a les competicions.
4. El treballador busca la competició que desitja mitjançant el seu nom o any de celebració i demana al sistema el formulari amb les seves dades.
5. El treballador demana el llistat dels judokes que participen a la competició i esborra als judokes que desitgi (només podrà esborrar als del seu club).
6. El sistema emmagatzema les noves dades.

Flux alternatiu d'esdeveniments:

1. El treballador del club s'autentica i accedeix al sistema.
 - 1.1. El sistema falla (en qualsevol moment).
 - 1.1.1. El sistema es reinicia demanant novament l'autenticació.
 - 1.2. L'autenticació és incorrecta.
 - 1.2.1. El sistema demana novament l'autenticació.
2. El sistema mostra el menú.
3. El treballador accedeix a l'apartat dedicat a les competicions.
4. El treballador busca la competició que desitja mitjançant el seu nom o any de celebració i demana al sistema el formulari amb les seves dades.
 - 4.1. La competició que busca el treballador no existeix.
 - 4.1.1. El sistema retorna un missatge d'error explicant la situació.
5. El treballador demana el llistat dels judokes que participen a la competició i esborra als judokes que desitgi (només podrà esborrar als del seu club).
6. El sistema emmagatzema les noves dades.
 - 6.1. Apareix un error al guardar les noves dades.
 - 6.1.1. El sistema retorna un missatge d'error explicant la situació.

3.4.14. Consultes

Cas d'ús: Consultes.

Objectius: Poder consultar diferents dades de clubs, Judokes i competicions.

Actors:

- Iniciador: Usuari genèric (pot ser treballador de la Federació, de club o un usuari no registrat)

Tipus: Primari.

Precondicions: -

Postcondicions: Les dades desitjades s'han presentat.

NOTA: Els usuaris genèrics, és a dir els no registrats, podran consultar dades sobre clubs, judokes i competicions, però només podran veure aquelles dades que estiguin autoritzats a visualitzar. Això exclou les dades personals dels judokes.

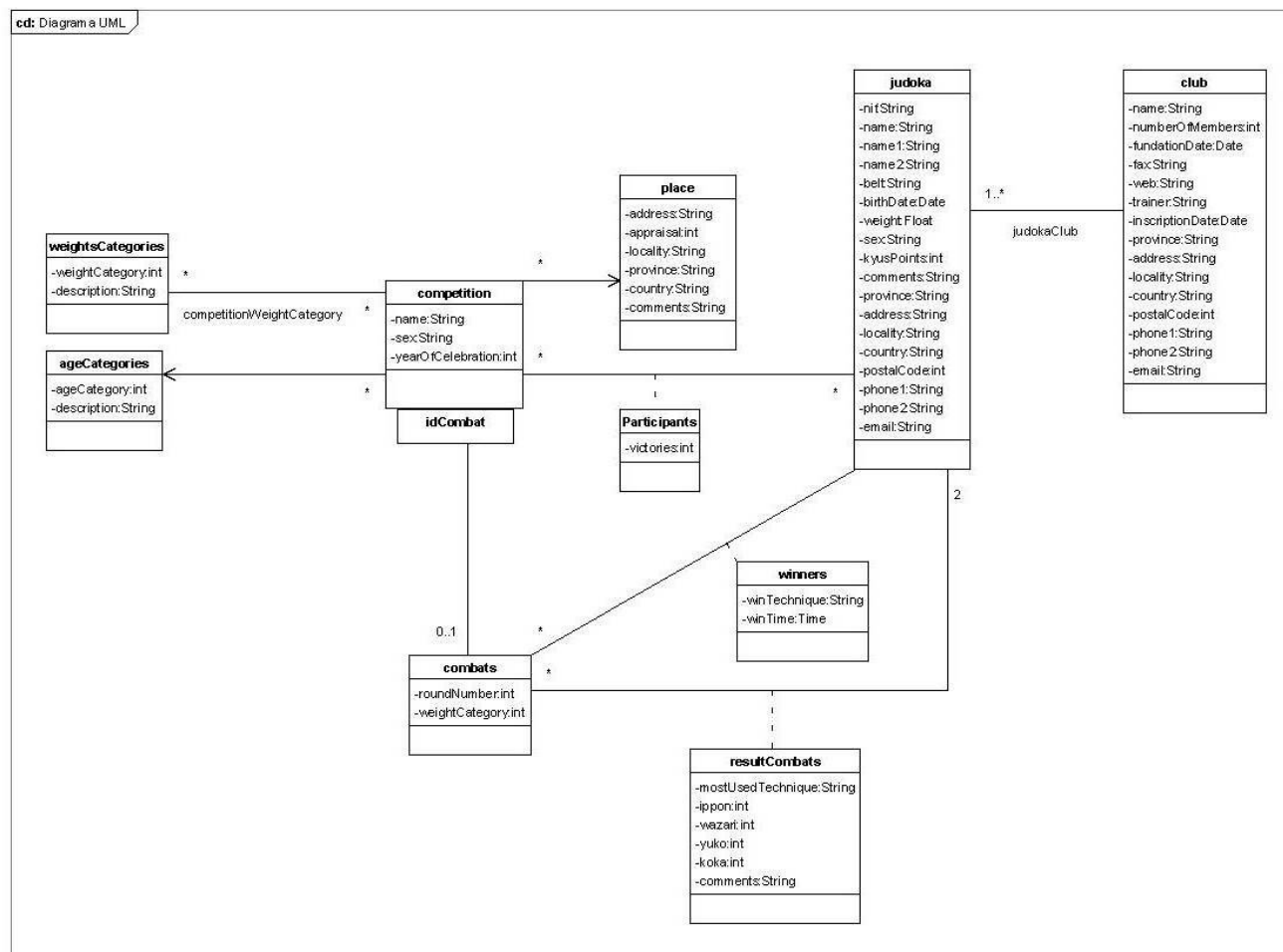
Flux principal d'esdeveniments:

1. L'usuari accedeix al sistema sense autenticar-se o fent-ho com a treballador de la Federació o com a treballador de club inscrit al sistema.
2. El sistema mostra el menú.
3. L'usuari accedeix a l'apartat que desitja consultar. Els possibles apartats són: judokes, clubs o competicions.
4. L'usuari busca la competició (amb el seu nom i/o any de celebració), el club (amb el seu nom) o el judoka (amb el seu cognom i/o DNI) i accedeix a les seves dades.
5. El sistema mostra les dades que l'usuari està permès a visualitzar.

Flux alternatiu d'esdeveniments:

1. L'usuari accedeix al sistema sense autenticar-se o fent-ho com a treballador de la Federació o com a treballador de club inscrit al sistema.
2. El sistema mostra el menú.
3. L'usuari accedeix a l'apartat que desitja consultar. Els possibles apartats són: judokes, clubs o competicions.
4. L'usuari busca la competició (amb el seu nom i/o any de celebració), el club (amb el seu nom) o el judoka (amb el seu cognom i/o DNI) i accedeix a les seves dades.
 - 4.1. L'usuari no troba la competició, el club o el judoka que buscava.
 - 4.1.1. El sistema mostra un missatge explicant que la competició, el club o el judoka buscat no existeix.
5. El sistema mostra les dades que l'usuari està permès a visualitzar.

3.5. Model de domini



Imatge 2. Model de domini

3.5.1. Explicació del model

A continuació s'explicaran les classes més importants vistes en el model de domini i la seva relació amb la resta (*Vegeu apartat 3.5*).

- Classe Competition: Les dades rellevants d'una competició són el seu nom, sexe dels competidors i any de celebració. Té una classe associada, *ageCategories*, que consisteix en el llistat de categories d'edats que contempla la competició.
- Classe judoka: Ve identificat pel seu DNI. Altres dades interessants d'un judoka són els seus noms, pes, cinturó, etc. A més a més d'un judoka també s'emmagatzemen altres dades que podríem anomenar "personals", com la seva direcció, telèfon, etc. Un Judoka pot participar a tantes competicions com desitgi. Així doncs, en cada competició que participi obtindrà uns resultats en cada combat que les classes *winners*, *combat* i *participantsCombats* s'encarregaran d'administrar. Un Judoka sempre pertanyerà a un club.
- Classe club: L'atribut identificador d'un club és el seu nom. A part del seu nom, els diferents atributs que ens interessin d'un club són el nombre de socis que té, quan es va fundar, l'entrenador de judo que té, etc. D'un club també ens poden interessar altres dades com la seva direcció, el seu telèfon, etc. que pertanyerien a la mateixa classe que les anomenades "dades personals" dels judokes.

4. ASPECTES DE DISSENY I IMPLEMENTACIÓ

Es poden presentar un gran nombre de problemes quan en una aplicació es barregen el codi d'accés a dades, el codi de la lògica de negoci i el codi de presentació. Aquestes aplicacions són complicades de mantenir, ja que qualsevol canvi afecta a tot el sistema. La barreja de codi en proporcions elevades dificulta o impossibilita la seva reutilització per causa de la dependència existent amb moltes altres classes. Afegir noves vistes de dades freqüentment requereix re-implementació de codi de la lògica de negoci. El codi d'accés a dades es susceptible al mateix problema.

El patró de disseny Model-Vista-Controlador resol aquest problema a través de la separació de l'accés a les dades, la lògica de negoci, la presentació de dades i la interacció amb l'usuari.

Per tant el patró MVC representa un mecanisme de millora de processos, fàcil de comprendre i aplicar (*Vegeu annex A, apartat A.1*).

4.1. Classes del disseny

A continuació s'explicarà la utilitat de les diferents capes del disseny enumerant les classes que pertanyen a cada nivell. Aquestes capes es presenten dividides segons si pertanyen al Model, la Vista o el Controlador del patró MVC.

4.1.1. Classes de Model

Les classes del model són les que defineixen la part més important d'una aplicació. Tenen com a principals feines modelitzar els conceptes de l'àmbit del negoci i establir les regles que cal tenir en compte a l'hora de tractar les dades.

JAVABEANS (ENTITATS)

Són les classes que formen el cor d'una aplicació i s'encarreguen de representar els seus conceptes, com per exemple un judoka o una competició.

L'especificació de JavaBeans de Sun Microsystems els defineix com "components de programari reutilitzables que es puguin manipular visualment en una eina de desenvolupament".

Per a funcionar com una classe JavaBean, una *objecte* classe ha d'obeir certes convencions sobre nomenclatura de mètodes, construcció, i comportament. Aquestes convencions permeten tenir eines que puguin utilitzar, reutilitzar, substituir, i connectar JavaBeans.

Capítol 4: Aspectes de disseny i implementació

Les convencions requerides són:

- ✓ La classe ha de ser serializable (capaç de salvar persistentment i de restablir el seu estat).
- ✓ Ha de tenir un constructor sense arguments.
- ✓ Les seves propietats han de ser accessibles mitjançant mètodes get i set que segueixen una convenció de nomenclatura estàndard.
- ✓ Ha de contenir determinats mètodes de maneig d'esdeveniments.

Les classes JavaBeans implementades en l'aplicació InfoJudo han estat les següents:

- **CompetitionObjectJB**: Implementada com una interfície. El que es pretén és tenir una classe genèrica que presenti els seus atributs en format de taula.
- **ClubJB**: Classe que modelitza un club de judo. Està formada pels atributs de l'entitat Club i implementa la interfície CompetitionObjectJB per facilitar la seva presentació en taules.
- **CombatsJB**: Classe que modelitza un combat de judo. Està formada pels atributs de l'entitat Combats i implementa la interfície CompetitionObjectJB per facilitar la seva presentació en taules.
- **CompetitionJB**: Classe que modelitza una competició de judo. Està formada pels atributs de l'entitat Competition i implementa la interfície CompetitionObjectJB per facilitar la seva presentació en taules.
- **JudokaJB**: Classe que modelitza un judoka de judo. Està formada pels atributs de l'entitat Judoka i implementa la interfície CompetitionObjectJB per facilitar la seva presentació en taules.
- **ResultCombatsJB**: Classe que modelitza els resultats d'un combat de judo. Està formada pels atributs de l'entitat ResultCombats i implementa la interfície CompetitionObjectJB per facilitar la seva presentació en taules.
- **WeightsCategoriesJB**: Classe que modelitza els pesos d'una competició de judo. Està formada pels atributs de l'entitat WeightsCategories i implementa la interfície CompetitionObjectJB per facilitar la seva presentació en taules.
- **AgeCategoriesJB**: Classe que modelitza les categories d'edats d'una competició de judo. Està formada pels atributs de l'entitat AgeCategories .
- **ParticipantsJB**: Classe que modelitza els participants d'una competició de judo. Està formada pels atributs de l'entitat Participants .
- **WinnersJB**: Classe que modelitza els guanyadors de combats en una competició de judo. Està formada pels atributs de l'entitat Winners .

DATA BASE BEANS

Les classes que pertanyen a aquest nivell del model són aquelles que proporcionen una interfície entre l'aplicació i les dades. En el cas de l'aplicació InfoJudo la font de les dades és una base de dades. Per tant, les *data base beans* s'encarreguen de connectar amb la base de dades i tractar la seva informació.

Les *data base beans* implementades per a l'aplicació InfoJudo són:

- DBAgeCategories: Classe que té accés directe a la taula AgeCategories de la base de dades. Obté les categories d'edats existents.
- DBClub: Classe que té accés directe a la taula Club de la base de dades. Crea, esborra, modifica i obté diverses dades de clubs.
- DBCombats: Classe que té accés directe a la taula Combats de la base de dades. Crea, esborra, modifica i obté diverses dades de combats.
- DBCompetition: Classe que té accés directe a la taula Competition de la base de dades. Crea, esborra, modifica i obté diverses dades de competicions.
- DBJudoka: Classe que té accés directe a la taula Judoka de la base de dades. Crea, esborra, modifica i obté diverses dades de judokes.
- DBParticipants: Classe que té accés directe a la taula Participants de la base de dades. Calcula el nombre de victòries d'un participant en una competició.
- DBResultCombats: Classe que té accés directe a la taula ResultCombats de la base de dades. Modifica els resultats d'un combat en una competició.
- DBWeightsCategories: Classe que té accés directe a la taula WeightsCategories de la base de dades. Afegeix i modifica categories de pesos d'una competició.
- DBWinners: Classe que té accés directe a la taula Winners de la base de dades. Calcula els guanyadors dels combats segons els seus resultats.
- DBLogin: Classe que té accés directe a la taula Login de la base de dades. Crea, esborra, modifica i comprova la correctesa de login's.
- DBConnection: Classe encarregada de retornar una connexió a la base de dades.

4.1.2. Classes de la Vista

La vista és el que retorna el model de domini per a que pugui ser visualitzada per l'usuari i permetre la interacció d'aquest usuari amb l'aplicació.

En Struts es retorna en forma de pàgina JSP (*Vegeu annex A, apartat A.3*). Un mateix element del domini pot tenir més d'una vista. Les pàgines JSP implementades en l'aplicació són aquestes:

Capítol 4: Aspectes de disseny i implementació

- Pàgines d'alta: AltaClub.jsp, AltaCompeticio.jsp, AltaJudoka.jsp i AltaWeightsCategories.jsp.
- Pàgines de consulta: ConsultarClub.jsp, ConsultarCombat.jsp, ConsultarCompeticio.jsp i ConsultarJudoka.jsp.
- Pàgines de modificació: ModificarClub.jsp, ModificarCombat.jsp, ModificarCompeticio.jsp i ModificarJudoka.jsp.
- Pàgines de presentació: ClubPresentation.jsp, CombatsPresentation.jsp, JudokaPresentation.jsp, CompetitionPresentation.jsp i WeightsPresentation.jsp.
- Pàgines d'errors: ErrorPermission.jsp i ErrorNoLogin.jsp.
- Altres: Login.jsp i Principal.jsp.

Aquestes pàgines utilitzen altres tecnologies com són JavaScript i CSS.

La connexió real entre les pàgines JSP i l'aplicació apareix quan s'introdueix el concepte de FormBeans.

FORMBEANS

Són les classes que condueixen les dades entre les vistes i l'aplicació. A grans trets el que farien és captar les dades d'entrada d'un formulari i transferir-les a les classes Action (*Vegeu apartat 4.2.3*). Al acabar l'execució de l'acció aquesta retorna el resultat novament a la vista per poder mostrar el resultat.

Els FormBeans emprats en l'aplicació són:

- AltaClubFB: Classe que crea el formulari necessari per donar d'alta un club.
- AltaCompeticioFB: Classe que crea el formulari necessari per donar d'alta una competició.
- AltaJudokaFB: Classe que crea el formulari necessari per donar d'alta un judoka.
- CombatFB: Classe que crea el formulari necessari per donar d'alta un combat.
- LoginFB: Classe que crea el formulari necessari per als login's.

4.1.3. Classes del Controlador

El controlador és l'encarregat d'administrar l'aplicació. Es qui comunica la vista amb el model. Rep els resultats del model i decideix quina es la següent vista que s'ha de mostrar. Les classes encarregades de fer tota aquesta feina són les anomenades Action (*Vegeu annex B, apartat B.3*).

Les classes Action implementades es poden classificar segons la feina que realitzen en l'aplicació.

- Alta:
 - AltaClubAction: Acció que dona d'alta un club en el sistema.
 - AltaCompeticioAction: Acció que dona d'alta una competició en el sistema.
 - AltaJudokaAction: Acció que dona d'alta un judoka en el sistema.
 - AltaCombatsAction: Acció que dona d'alta un combat d'una competició en el sistema.
 - AltaParticipant: Acció que dona d'alta un participant a una competició en el sistema.

Capítol 4: Aspectes de disseny i implementació

- Esborrar:
 - DeleteClubAction: Acció que esborra un club del sistema.
 - DeleteCompetitionAction: Acció que esborra una competició del sistema amb tots els seus combats, guanyadors i participants.
 - DeleteJudokaAction: Acció que esborra un judoka del sistema.
 - DeleteCompetitionParticipantAction: Acció que esborra un participant a una competició del sistema.
- Cercar:
 - FindClubAction: Acció que busca un o més clubs a partir de les dades que proporciona l'usuari.
 - FindCompetitionAction: Acció que busca una o més competicions a partir de les dades que proporciona l'usuari.
 - FindJudokaAction: Acció que busca un o més judokes a partir de les dades que proporciona l'usuari.
 - FindCombatsAction: Acció que busca un o més combats d'una competició a partir de certes dades.
- Modificar:
 - ModifyClubAction: Acció encarregada de modificar un club.
 - ModifyCompetitionAction: Acció encarregada de modificar una competició.
 - ModifyJudokaAction: Acció encarregada de modificar un judoka.
 - ModifyCombatsAction: Acció encarregada de modificar un combat d'una competició.
- Presentar:
 - HTMLCodeCreator: Encarregat de generar el codi HTML dinàmic per la presentació de dades en taules.
 - CompetitionObjectPresentationAction: Acció que actua de superclasse establint els passos generals necessaris per presentar en taules diferents entitats.
 - ClubPresentationAction: Subclasse de CompetitionObjectPresentationAction que defineix una acció per presentar clubs en taules.
 - CompetitionPresentationAction: Subclasse de CompetitionObjectPresentationAction que defineix una acció per presentar competicions en taules.
 - JudokaPresentationAction: Subclasse de CompetitionObjectPresentationAction que defineix una acció per presentar judokes en taules.
 - WeightsPresentationAction: Subclasse de CompetitionObjectPresentationAction que defineix una acció per presentar categories de pesos d'una competició en taules.
 - CombatsPresentationAction: Subclasse de CompetitionObjectPresentationAction que defineix una acció per presentar combats d'una competició en taules.

4.2. Disseny de la base de dades

Per poder emmagatzema tota la informació necessària, l'aplicació utilitza una base de dades, dissenyada a partir del model de domini vist (*Vegeu apartat 3.5*).

El seu model de dades relacional és el següent:

AgeCategories (AgeCategory, description)

Competition (id, name, sex, yearOfCelebration, address, appraisal, locality, province, country, comments, *ageCategory*)

Combats (IdCombat, *id*, roundNumber, weightCategory)

Participants (*id*, *nif*, victories)

ResultCombats (*idCombat*, *id*, *nif*, MostUsedTechnique, ippon, wazari, yuko, koka, comments)

Club (Name, numberOfMembers, foundationDate, fax, web, trainer, inscriptionDate, address, province, locality, country, postalCode, phone1)

Judoka (*nif*, name, name1, name2, belt, birthDate, weight, sex, comments, address, province, locality, country, postalCode, phone1)

Winners (*idCombat*, *id*, *nif*, winTechnique, winTime)

Els atributs subratllats són els identificadors mentre que els que estan en cursiva són els atributs foranis.

5. CONCLUSIONS I TREBALL FUTUR

Per últim amb aquest capítol arribem a la fi del nostre viatge.

En la introducció d'aquest projecte parlava del somni d'un nen aficionat a la informàtica, el somni d'una aplicació que ajudés a la Federació lleidatana de judo i difongués aquest esport. Puc ben dir, sense por a equivocar-me que aquest somni s'ha complert.

La nova aplicació InfoJudo redueix a 0 tot el treball organitzatiu que s'hauria de portar a terme a mà en esdeveniments esportius de judo.

Calcula lligues i enfrontaments tenint en compte edat i pes dels participants, tot reduint la ineficiència que comportaria que aquesta feina l'hagués de fer una persona.

Porta un control exhaustiu de clubs i judokes poden accedir a tota aquesta informació en qualsevol moment, en qualsevol lloc del món.

Qualsevol club o judoka pot saber quines competicions es realitzaran i apuntar-se sense tenir que tramitar absolutament res més.

A demés InfoJudo té una segona vessant, que a mi personalment hem sembla més interessant si cap, la difusió del judo. Això vol dir que no només un judoka pot mirar qui són els seus rivals a batre, qui va guanyar l'any passat el campionat que desitja emportar-se o quin és el club més important, sinó que això ho pot fer qualsevol aficionat.

No cal dir que encara queda molta feina per fer. Tant com a usuari i com a programador sóc conscient de que InfoJudo és encara una aplicació molt jove.

El punt més feble de l'aplicació en l'aspecte de programació seria el seu control d'errors. El següent punt en el desenvolupament futur d'InfoJudo seria millorar aquest control d'errors. Com a usuari sorgeixen moltes idees com la incorporació d'estadístiques de clubs i judokes, entrevistes als protagonistes i moltes coses més.

La possibilitat de que en un futur la col·laboració entre aquesta aplicació i la Federació lleidatana de judo sigui oficial m'impulsa a seguir treballant. Els suggeriments i peticions de la Federació seran escoltades i segur que ajuden a millorar molt més InfoJudo.

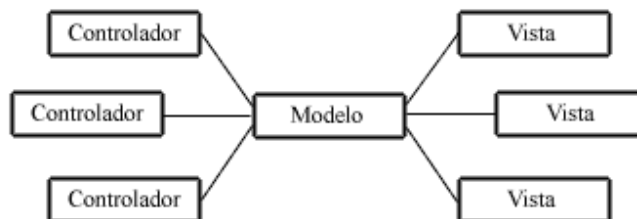
Es podria dir que el somni inicial amb el que obríem el projecte, al igual que la matèria, no s'ha destruït sinó que s'ha transformat. Ara somio en convertir aquesta aplicació en un punt de referència tant per treballadors, com per judokes i aficionats.

Annex

A**A. TECNOLOGIES WEB****A.1. ARQUITECTURA Model/Vista/Controlador**

L'arquitectura MVC (Model/View/Controller) va ser introduïda com a part de la versió Smalltalk-80 del llenguatge de programació Smalltalk. Va ser dissenyada per reduir l'esforç de programació necessari en la implementació de sistemes múltiples i sincronitzats de les mateixes dades. Les seves característiques principals són que el Model, les Vistes i els Controladors es tracten com entitats separades; això fa que qualsevol canvi produït en el Model quedi reflectit automàticament en cadascuna de les Vistes.

En la figura següent, es pot observar l'arquitectura MVC de forma general. Existeix un Model, múltiples Controladors que manipulen el Model, y diverses Vistes de les dades del Model, que canvien quan canvia l'estat d'aquest.



Imatge 3. Esquema Model-Vista-Controlador

Aquesta arquitectura presenta diverses avantatges:

- La separació entre els components d'un programa és clara, d'aquesta manera es permet implementar-los individualment.
- Existeix un API molt ben definit. Qualsevol que utilitzi l'API podrà substituir el Model, la Vista o el Controlador sense dificultat.
- La connexió entre el Model i les seves Vistes és dinàmica, es a dir, es produeix en temps d'execució i no en temps de compilació.

Al incorporar l'arquitectura MVC en un disseny, les peces d'un programa es poden construir de forma separada i després unir-les en temps d'execució. Si un dels Components, posteriorment no funciona com s'esperava, es pot reemplaçar sense que afecti a les altres peces.

A.1.1. Definició de les parts

- **Model:** És l'objecte que representa les dades del programa. Manipula les dades i controla totes les seves transformacions. No té coneixement específic dels Controladors o de les seves Vistes, ni tan sols conté referència a ells. És el propi sistema el que té encomanada la responsabilitat de mantenir enllaços entre el Model i les seves Vistes i notificar a les Vistes quan canvia el Model.
- **Vista:** És l'objecte que administra la presentació visual de les dades representats pel Model. Genera una representació visual del Model i mostra les dades de l'usuari. Interactua amb el Model a través d'una referència a ell.
- **Controlador:** És l'objecte que proporciona significat a les ordres de l'usuari, actuant sobre les dades representades pel Model. Quan es realitza algun canvi, entra en acció, ja sigui per canvis en informació del Model o per alteracions de la Vista.

A.2. HTML

HTML (Acrònim de *Hyper Text Markup Language*, en català, "llenguatge de marcat d'hipertext"), és un llenguatge de marcat que deriva de l'SGML dissenyat per estructurar textos i relacionar-los en forma d'hipertext. Gràcies a Internet i als navegadors web, s'ha convertit en un dels formats més populars que existeixen per a la construcció de documents.

Les etiquetes bàsiques d'HTML, d'obligada presència en tot document són:

- **<!DOCTYPE>:** És l'etiqueta que permet definir el tipus de document HTML que s'empra. Existeixen tres tipus bàsics: l'estricta (Strict), el transicional (Transitional) i el de marcs (Frameset).
- **<html>:** És l'etiqueta arrel de qualsevol document HTML o XHTML.
- **<head>:** Defineix la capçalera del document HTML. Permet declarar metainformació del document que no es mostra directament en el navegador. Aquesta informació és d'especial rellevància pels indexadors i cercadors automàtics.
- **<body>:** Defineix el cos del document. Aquesta és la part del document HTML que es mostra en el navegador.

Dintre de la capçalera <HEAD> hi podem trobar:

- **<title>:** Permet definir el títol de la pàgina. En navegadors gràfics el contingut del title apareix a la barra del títol a sobre de la finestra.
- **<meta>:** Permet definir metainformacions del document tals com l'autor, la data de realització, la codificació del document (UTF, ISO, etc.), les paraules clau i la descripció del mateix
- **<LINK>:** Permet definir metadades complementàries a les del meta tals com el document anterior, el següent, el capítol al qual pertany el document, la pàgina glossari, etc.

Dintre del cos <BODY> hi podem trobar:

- <a>: Etiqueta ancla. Crea un enllaç a un altre document o a una altra zona del mateix, segons els atributs.
- <h1>, <h2>, ... <h6>: capçaleres o títols del document, acostumen a distingir-se per mida.
- <div>: Divisió estructural de la pàgina.
- <p>: Paràgraf.
-
: Salt de línia.
- <table>: Indica el començament d'una taula, després s'haurà de definir les files amb <tr> i les cel·les dintre de les files amb <td>.
- : Llista desordenada (sense numerar). Els ítems es defineixen amb .
- : Llista ordenada (numerat). Els ítems es defineixen amb .
- <dl>: Llista de definició. Hi ha dos tipus d'ítem; el dt i el dd.
- <dt>: Terme a definir.
- <dd>: Definició del terme.

Excepte unes poques etiquetes, la majoria requereixen ser tancades escrivint la mateixa etiqueta precedida d'una barra "/". Exemple: <html>...</html>

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
    <html lang="es">
        <head>
            <title>Exemple</title>
        </head>
        <body>
            <p>exemple</p>
        </body>
    </html>
```

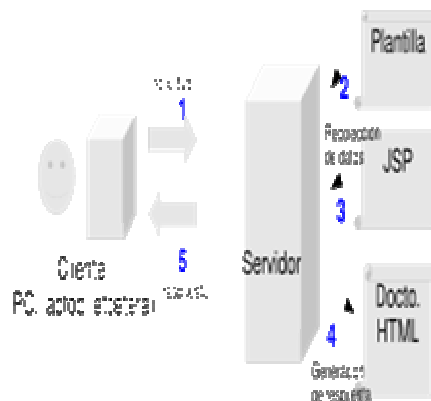
A.3. JSP

Java Server Pages (JSP) és una tecnologia Java que permet generar contingut dinàmic per a web, en forma de documents HTML, XML o de altres tipus.

Les JSP's permeten la utilització de codi Java mitjançant scripts. A més és possible utilitzar alguna acció JSP predefinida a través d'etiquetes. Aquestes etiquetes poden ser enriquides mitjançant la utilització de llibreries d'etiquetes (TagLibs o Tag libraries) externes o inclús personalitzades.

El funcionament general de la tecnologia JSP és que el Servidor d'Aplicacions interpreta el codi contingut en la pàgina JSP per construir un Servlet, la sortida del qual serà un document estàtic (típicament HTML) que es presentarà en la pantalla del Navegador de l'usuari.

JSP -> Servidor Aplicacions (Servlets) -> Clients (Navegador)



Imatge 4. Funcionament general tecnologia JSP

JSP no es pot considerar un script al 100%, ja que abans d'executar-se el Servidor d'Aplicacions compila el contingut del document JSP (script i etiquetes) i genera una classe Servlet, per tant, es pot dir que encara que aquest procés sigui transparent per al programador no deixa de ser una tecnologia compilada.

El principal avantatge de JSP en front d'altres llenguatges és que permet integrar-se amb classes Java(.class) el que permet separar en nivells les aplicacions web, emmagatzemant en classes Java les parts que consumeixen més recursos (així com les que requereixen major seguretat) i deixant la part encarregada de formatejar el document HTML en l'arxiu JSP. La idea fonamental d'aquest criteri és el de separar la lògica del negoci de la presentació de la informació.

A.4. CSS

Els fulls d'estil en cascada (*Cascading Style Sheets*, CSS) són un llenguatge formal usat per a definir la presentació d'un document estructurat escrit en HTML o XML (i per extensió en XHTML). El W3C (World Wide Web Consortium) és l'encarregat de formular l'especificació dels fulls d'estil que servirà d'estàndard per als agents d'usuari o navegadors.

La idea que es troba darrere del desenvolupament de CSS és separar l'*estructura* d'un document de la seva *presentació*.

Per exemple, l'element d'HTML <H1> indica que un bloc de text és un encapçalament i que és més important que un bloc etiquetat com <H2>. Versions més antigues d'HTML permetien atributs extra dins l'etiqueta oberta per donar-li format (com el color o la mida de font). No obstant això, cada etiqueta havia de disposar d'aquesta informació si es desitjava un disseny consistent per a una pàgina, i a més, una persona que llegís aquesta pàgina amb un navegador, perdia totalment el control sobre la visualització del text.

Quan s'utilitza CSS, l'etiqueta <H1> no hauria de proporcionar informació sobre com serà visualitzat, només marca l'estructura del document. La informació d'estil separada en una fulla d'estil específica com s'ha de mostrar <H1>: color, font, alineació del text, grandària, i altres característiques no visuals com definir el volum d'un sintetitzador de veu, per exemple.

La informació d'estil pot ser adjuntada tant com un document separat o en el mateix document HTML. En aquest document, podrien definir-se estils generals en la capçalera del document o en cada etiqueta particular mitjançant l'atribut "style".

Els avantatges d'utilitzar CSS (o altre llenguatge d'estil) són:

- Control centralitzat de la presentació d'un lloc web complet amb el que s'agilita de forma considerable l'actualització del mateix.
- Els navegadors permeten als usuaris especificar el seu propi full d'estil local que serà aplicada a un lloc web remot, amb el que augmenta considerablement l'accessibilitat. Per exemple, persones amb deficiències visuals poden configurar el seu propi full d'estil per a augmentar la grandària del text o remarcar més els enllaços.
- Una pàgina pot disposar de diferents fulles d'estil segons el dispositiu que la mostri o fins i tot a elecció de l'usuari. Per exemple, per a ser impresa, mostrada en un dispositiu mòbil, o ser "llegida" per un sintetitzador de veu.
- El document HTML en si mateix és més clar d'entendre i s'aconsegueix reduir considerablement la seva grandària.

A.5. JAVASCRIPT

JavaScript es un llenguatge script basat en el concepte de prototip, implementat originàriament per Netscape Communications Corporation, i que va derivar en l'estandard ECMAScript. És conegut sobretot pel seu ús en pàgines web, però també s'utilitza en altres aplicacions.

Al contrari que Java, JavaScript no és un llenguatge orientat a objectes pròpiament dit, ja que no disposa d'herència, sinó que és un llenguatge basat en prototips, ja que les noves classes es generen clonant les classes base (prototips).

Tradicionalment, s'utilitzava en pàgines web HTML, per realitzar feines i operacions en el marc de l'aplicació únicament client, sense accés a funcions del servidor. JavaScript s'executa en l'agent de l'usuari al mateix temps que les sentències van descarregant-se junt amb el codi HTML.

Per indicar-li a una pàgina HTML que s'utilitzarà llenguatge JavaScript s'ha d'escriure la següent etiqueta:

```
<script type="text/javascript">  
    // codi JavaScript  
</script>
```

Característiques tècniques de JavaScript.

- Interpretat (no compilat) des de el navegador del client.
- Enllaçat dinàmic. Els objectes als que es fa referència han d'existir en temps d'execució (llenguatge interpretat).
- No requereix la presència de servidor, ja que manté un funcionament local.
- L'aplicació resideix en el client.
- El codi està integrat en el HTML. Queda ocult a l'usuari , però pot formar part del mateix arxiu.
- No és necessari declarar el tipus de variables.

Annex

B

B. STRUTS

Struts és un framework d'aplicació, de codi font obert, escrit amb Java, que simplifica el desenvolupament d'aplicacions web. Implementa el patró de disseny MVC. És compatible amb la plataforma J2EE i bàsicament està construït sobre les tecnologies de Servlets i JSP.. Struts uneix aquestes tecnologies i els recursos de l'aplicació en un únic framework, ocultant al programador els detalls.

Va ser creat per Craig McClanahan i fins fa pocs mesos va formar part del subprojecte Jakarta, del Apache Software Foundation.

Struts amaga al programador els detalls de les tecnologies que utilitza com HTTP, Servlets, JSP, etc.

B.1. STRUTS I MVC

Com ja s'ha comentat, Struts implementa el patró MVC (Model View Controller). L'aplicació per tant es divideix en tres subsistemes:

- Model: representa l'estat de les dades de l'aplicació.
- Vista: És la interfície amb l'usuari, es a dir, les pàgines amb les que interacciona l'usuari (JSP, HTML).
- Controlador: Implementa el flux de l'aplicació.

Struts proveeix de components les tres capes de l'arquitectura:

- La capa del Model s'implementa habitualment amb JavaBeans o EJB.
- La capa del Controlador s'implementa amb un conjunt d'Actions i mapejades en un arxiu anomenat struts-config.xml.
- La capa de la Vista la formen les diferents pàgines en format JSP que obtenen suport de les llibreries (custom tags).

B.2. DIAGRAMA DE SEQÜÈNCIA

Pas a pas Struts portaria a terme les següents accions (diagrama de seqüència):

- 1) L'usuari clika sobre un link d'una pàgina HTML o envia un formulari mitjançant un botó submit.
- 2) L'Action Servlet rep el requeriment i busca informació de mapeig en l'arxiu struts-config.xml que utilitzarà per enrutar el requeriment a l'objecte Action corresponent.
- 3) El framework invoca el mètode execute() de l'objecte Action apropiat, funció principal del qual es processar el requeriment entrant i retornar un objecte ActionForward que identifica a on direccionar el flux (per exemple a una JSP o a un altre objecte Action).
- 4) El requeriment es processat invocant a funcions apropiades del Model.
- 5) El component del Model invocat retorna a l'Action.
- 6) L'objecte ActionForward retornat per l'Action es captat per l'ActionServlet qui l'utilitza per seleccionar el pròxim recurs de l'aplicació a executar, habitualment una JSP que mostra la resposta.
- 7) La JSP es invocada per l'ActionServlet i retorna al navegador una pàgina HTML.
- 8) Es presenta a l'usuari una nova pàgina al navegador.

B.3. COMPONENTS DE LA CAPA CONTROLADORA

Com s'ha comentat anteriorment, Struts subministra de components les diferents capes de l'arquitectura MVC. En aquest apartat es tracta la capa més important d'aquesta arquitectura, la capa controladora.

Els components més importants que Struts proporciona a la capa són els següents:

- ActionServlet: És la “columna vertebral” del framework Struts. Com tots els Servlets, viu al contenidor Web. La seva funció principal és dirigir el flux de control de l'aplicació.
- Action: És l'objecte que s'encarrega dels requeriments HTTP. Pot fer tot el que fa un Servlet, però no és un Servlet. Interactua amb el model.
- ActionForwards: Representa les URI's de l'aplicació. Habitualment es defineix en l'arxiu de configuració struts-config.xml. Tenen un nom lògic i un path.
- ActionMappings: Proveeixen als objectes Action d'una URI o path. Funciona com una “fulla de ruta” per a l'ActionServlet.
- ActionForms: Són JavaBeans amb un parell de mètodes addicionals que ofereixen la possibilitat de validar dades d'entrada i resetejar-les. Administren de forma automàtica els inputs dels formularis HTML's.
- L'arxiu de configuració struts-config.xml: S'utilitza per carregar els components crítics del framework. L'arxiu és a Struts el que és el web.xml per al Contenedor Web. L'ActionServlet llegeix l'arxiu de configuració per crear i configurar els objectes que el framework necessita.

Annex

C

C. ESPECIFICACIÓ

C.1. Operacions De La Classe DBCompetition**public void createCompetition(CompetitionJB cJB);**

- **Crida:** DBc.createCompetition(cJB);
- **Precondició:** -
- **Postcondició:**

Si la competició *cJB* ja existeix a la BD es llença l'excepció *CompetitionExistsException* i *BD=BD'*

Sino *BD=BD'* Unió *cJB* (emmagatzema la competició *cJB* a la BD).

public void addCompetitionWeightCategory (String weightCategory, CompetitionJB cJB);

- **Crida:** DBc.addCompetitionWeightCategory(weightCategory, cJB);
- **Precondició:** -
- **Postcondició:**

Si la competició *cJB* no existeix a la BD es llença l'excepció *CompetitionNotExistsException* i *BD=BD'*

Sino Si la categoria de pesos *weightCategory* no existeix a la BD es llença l'excepció *WeightCategoryNotExistsException* i *BD=BD'*

Sino *weightCategory* s'afegeix com a nova categoria de pesos de la competició *cJB*.

public void addParticipants (String nif, CompetitionJB cJB);

- **Crida:** DBc.addParticipants (nif, cJB);
- **Precondició:** -
- **Postcondició:**

Si la competició *cJB* no existeix a la BD es llença l'excepció *CompetitionNotExistsException* i *BD=BD'*

Sino Si el judoka identificat per *nif* no existeix a la BD es llença l'excepció *JudokaNotExistsException* i *BD=BD'*

Sino el judoka identificar per *nif* s'afegeix com a participant de la competició *cJB*.

public void deleteCompetition (CompetitionJB cJB);

- **Crida:** DBc.deleteCompetition(cJB);
- **Precondició:** -
- **Postcondició:**

Si la competició *cJB* no existeix a la BD es llença l'excepció *CompetitionNotExistsException* i *BD=BD'*

Sino *BD=BD'* - *cJB* (esborra la competició *cJB* de la BD).

public void deleteParticipants (String nif, CompetitionJB cJB);

- **Crida:** DBc.deleteParticipants (nif, cJB);
- **Precondició:** -
- **Postcondició:**

Si la competició *cJB* no existeix a la BD es llença l'excepció *CompetitionNotExistsException* i *BD=BD'*

Sino Si el judoka identificat per *nif* no existeix a la BD es llença l'excepció *JudokaNotExistsException* i *BD=BD'*

Sino el judoka identificar per *nif* s'esborra com a participant de la competició *cJB*.

public void modifyCompetition (CompetitionJB cJB);

- **Crida:** DBc.modifyCompetition(cJB);
- **Precondició:** -
- **Postcondició:**

Si la competició *cJB* no existeix es llença l'excepció *CompetitionNotExistsException* i *BD=BD'*

Sino la competició *cJB* de la BD és modificada amb els nous valors de *cJB*.

Protected String findCompetitionId (String name, String sex, String age, String year);

- **Crida:** *id* = DBc.findCompetitionId (name, sex, age, year);
- **Precondició:** -
- **Postcondició:**

Si la competició *cJB* no existeix es llença l'excepció *CompetitionNotExistsException* i *BD=BD'*

Sino retorna el id de la competició i *BD=BD'*.

public CompetitionJB findCompetition (String id);

- **Crida:** **C** = DBc.findCompetition (name, sex, age, year);
- **Precondició:** -
- **Postcondició:**

Si la competició identificada per id no existeix es llença l'excepció CompetitionNotExistsException i BD=BD'

Sino retorna la competició identificada per id i BD=BD'.

**public ArrayList<JudokaJB> findParticipantsByCompetition
(CompetitionJB cJB);**

- **Crida:** L = DBc.findParticipantsByCompetition (cJB);
- **Precondició:** -
- **Postcondició:**

Si la competició *cJB* no existeix es llença l'excepció CompetitionNotExistsException i BD=BD'

Sino *L* és igual als participants de la competició *cJB* i BD=BD'.

**public ArrayList<JudokaJB> findParticipantsByCompetitionAndWeight
(String categoryWeight, CompetitionJB cJB);**

- **Crida:** L = DBc.findParticipantsByCompetitionAndWeight(categoryWeight, cJB);
- **Precondició:** -
- **Postcondició:**

Si la competició *cJB* no existeix a la BD es llença l'excepció CompetitionNotExistsException i BD=BD'

Sino Si la categoria de pes identificada per *categoryWeight* no existeix a la BD es llença l'excepció WeightCategoryNotExistsException i BD=BD'

Sino *L* és igual als participants de la competició *cJB* que són de la categoria de pes *categoryWeight* i BD=BD'.

**public ArrayList<WeightsCategoriesJB>
findCompetitionWeightsCategories (CompetitionJB cJB);**

- **Crida:** L = DBc.findCompetitionWeightsCategories (cJB);
- **Precondició:** -
- **Postcondició:**

Si la competició *cJB* no existeix es llença l'excepció CompetitionNotExistsException i BD=BD'

Sino *L* és igual a les categories de pesos de la competició *cJB* i BD=BD'.

public ArrayList<CompetitionJB> getCompetitions ();

- **Crida:** L = DBc.getCompetitions();
- **Precondició:** -
- **Postcondició:**

L és una ArrayList d'objectes CompetitionJB amb totes les competicions existents a la base de dades i BD=BD'.

public ArrayList<CompetitionJB> searchCompetitions (HashMap<String, String> hash);

- **Crida:** L = DBc.searchCompetitions(hash);
- **Precondició:** -
- **Postcondició:**

L és una ArrayList d'objectes CompetitionJB obtinguts a partir del paràmetre hash i BD=BD'.

public ArrayList<CompetitionJB> findCompetitionsByYearOfCelebration (String yearOfCelebration);

- **Crida:** L = DBc.findCompetitionsByYearOfCelebration (yearOfCelebration);
- **Precondició:** -
- **Postcondició:**

L és una ArrayList d'objectes CompetitionJB que tenen el mateix valor de paràmetre que yearOfCelebration i BD=BD'.

public ArrayList<CombatsJB> findCombatsByCompetition(String id);

- **Crida:** L = DBc.findCombatsByCompetition (id);
- **Precondició:** -
- **Postcondició:**

Si la competició identificada per id no existeix a la BD es llença l'excepció CompetitionNotExistsException i BD=BD'.

Sino Si L és igual als combats de la competició identificada per id i BD=BD'.

public ArrayList<JudokaJB> calculateCompetitionClasification (CompetitionJB compjb);

- **Crida:** L = DBc.calculateCompetitionClasification (compjb);
- **Precondició:** -
- **Postcondició:**

Si la competició compjb no existeix a la BD es llença l'excepció CompetitionNotExistsException i BD=BD'.

Sino L és una ArrayList de judokesJB ordenada segons els seus resultats als combats i BD=BD'.

C.2. Operacions De La Classe DBJudoka

public void addJudoka(JudokaJB judoka1, String clubName);

- **Crida:** DBj.addJudoka(judoka1, clubName);
- **Precondició:** -
- **Postcondició:**

Si el club identificar per *clubName* no existeix a la BD es llença l'excepció ClubNotExistsException i BD=BD'

Sino Si el judoka *judoka1* ja existeix a la BD es llença l'excepció JudokaExistsException i BD=BD'

Sino BD=BD' Unió judoka1 (emmagatzema el judoka *judoka1* a la BD).

public void modifyJudoka (JudokaJB judoka1);

- **Crida:** DBj.modifyJudoka(judoka1);
- **Precondició:** -
- **Postcondició:**

Si el judoka *judoka1* no existeix es llença l'excepció JudokaNotExistsException i BD=BD'

Sino el judoka *judoka1* de la BD és modificat amb els nous valors de *judoka1*.

public JudokaJB findJudokaByNif (String nif);

- **Crida:** J = DBj.findJudokaByNif (nif);
- **Precondició:** -
- **Postcondició:**

Si el judoka identificar per *nif* no existeix es llença l'excepció JudokaNotExistsException i BD=BD'

Sino J és igual al judoka identificat per *nif* i BD=BD'.

public ClubJB findJudokaClub (String nif);

- **Crida:** J = DBj.findJudokaClub (nif);
- **Precondició:** -
- **Postcondició:**

Si el judoka identificat per *nif* no existeix es llença l'excepció JudokaNotExistsException i BD=BD'

Sino J és igual al club del judoka identificat per *nif* i BD=BD'.

public void modifyJudokaClub (JudokaJB judoka, ClubJB club1);

- **Crida:** DBj.modifyJudokaClub (judoka1, club1);
- **Precondició:** -
- **Postcondició:**

Si el judoka *judoka1* no existeix es llença l' excepció JudokaNotExistsException i BD=BD'

Sino Si el club no existeix a la BD es llença l' excepció ClubNotExistsException i BD=BD'

Sino el nou club del judoka *judoka1* és *club1*.

public void deleteJudoka (JudokaJB cJB);

- **Crida:** DBc.deleteJudoka(cJB);
- **Precondició:** -
- **Postcondició:**

Si la competició *cJB* no existeix a la BD es llença l' excepció JudokaNotExistsException i BD=BD'

Sino BD=BD' - cJB (esborra el judoka *cJB* de la BD).

public ArrayList<JudokaJB> searchJudokas (HashMap<String, String> hash, String actionp);

- **Crida:** L = DBc.searchJudokas(hash, actionp);
- **Precondició:** -
- **Postcondició:**

L és una ArrayList d'objectes JudokaJB obtinguts a partir del paràmetre hash i actionp. A demés BD=BD'.

public ArrayList<JudokaJB> getJudokas ();

- **Crida:** L = DBc.getJudokas();
- **Precondició:** -
- **Postcondició:**

L és una ArrayList d'objectes JudokaJB amb tots els judokes existents a la base de dades i BD=BD'.

C.3. Operacions De La Classe DBClub

public void createClub(ClubJB club1);

- **Crida:** DBc.createClub(club1);
- **Precondició:** -
- **Postcondició:**

Si el club *club1* ja existeix a la BD es llença l'excepció ClubExistsException i BD=BD'

Sino BD=BD' Unió club1 (emmagatzema el club *club1* a la BD).

public void modifyClub (ClubJB club1);

- **Crida:** DBc.modifyClub(club1);
- **Precondició:** -
- **Postcondició:**

Si el club *club1* no existeix es llença l'excepció ClubNotExistsException i BD=BD'

Sino el club *club1* de la BD és modificat amb els nous valors de *club1*.

public void deleteClub (ClubJB club1);

- **Crida:** DBc.deleteClub(club1);
- **Precondició:** -
- **Postcondició:**

Si el club *club1* no existeix a la BD es llença l'excepció ClubNotExistsException i BD=BD'

Sino si el club club1 té algún judoka assignat es llença l'excepció ClubNotDeleteException

Sino BD=BD' - club1 (esborra el club *club1* de la BD).

public ClubJB findClubByName (String name);

- **Crida:** C = DBc.findClubByName (name);
- **Precondició:** -
- **Postcondició:**

Si el club identificar per *name* no existeix es llença l'excepció ClubNotExistsException i BD=BD'

Sino C és igual al club identificat per *name* i BD=BD'.

public ArrayList<JudokaJB> findJudokaByClub (ClubJB club1);

- **Crida:** L = DBc.findJudokaByClub (club1);
- **Precondició:** -
- **Postcondició:**

Si el club *club1* no existeix es llença l'excepció ClubNotExistsException i BD=BD'

Sino *L* és igual als judokes del club *club1* i BD=BD'.

public ArrayList<ClubJB> getClubs ();

- **Crida:** L = DBc.getClubs();
- **Precondició:** -
- **Postcondició:**

L és una ArrayList d'objectes ClubsJB amb tots els clubs existents a la base de dades i BD=BD'.

public ArrayList<ClubJB> searchClubs (HashMap<String, String> hash);

- **Crida:** L = DBc.searchClubs(hash);
- **Precondició:** -
- **Postcondició:**

L és una ArrayList d'objectes ClubJB obtinguts a partir del paràmetre hash i BD=BD'.

C.4. Operacions De La Classe DBResultcombats

public void modifyResultCombats (ResultCombatsJB result, String idCombat, String nif, CompetitionJB cJB);

- **Crida:** DBrc.modifyResultCombats(result, idCombat, nif, cJB);
- **Precondició:** -
- **Postcondició:**

Si la competició *cJB* no existeix a la BD es llença l'excepció CompetitionNotExistsException i BD=BD'

Sino Si el judoka identificar per *nif* no existeix es llença l'excepció JudokaNotExistsException i BD=BD'

Sino Si el combat identificat per *idCombat* no existeix es llença l'excepció CombatNotExistsException i BD=BD'

Sino *result* passa a ser el nou resultat del combat identificat per *idCombat* del judoka identificat per *nif* a la competició *cJB*.

C.5. Operacions De La Classe DBCombats

public void createCombats (CompetitionJB cJB);

- **Crida:** DBc.createCombats(cJB);
- **Precondició:** -
- **Postcondició:**

Si la competició *cJB* no existeix a la BD es llença l'excepció *CompetitionNotExistsException* i *BD=BD'*

Sino es calculen i emmagatzement els emparellaments dels combats de la competició *cJB* segons el seu tipus i les seves categories de pesos.

public ArrayList<JudokaJB> findParticipantCombatsByCompetition (CompetitionJB compjb, int idCombat);

- **Crida:** *L* = DBc. findParticipantCombatsByCompetition (compjb, idCombat);
- **Precondició:** -
- **Postcondició:**

Si la competició *compjb* no existeix es llença l'excepció *CompetitionNotExistsException* i *BD=BD'*

Sino *L* és igual als judokes participants de la competició *compjb* i *BD=BD'*.

public CombatsJB findCombatsByCompetitionAndJudokas(CompetitionJB cJB, String nif1, String nif2);

- **Crida:** *L* = DBc. findCombatsByCompetitionAndJudokas (cJB, nif1, nif2);
- **Precondició:** -
- **Postcondició:**

Si la competició *compjb* no existeix es llença l'excepció *CompetitionNotExistsException* i *BD=BD'*

Sino *L* és igual combats dels combats dels judokes *nif1* i *nif2* han fet junts en la competició *compjb* i *BD=BD'*.

public ArrayList<ResultCombatsJB> findResultCombatsByCompetitionAndId(CompetitionJB cJB, int idCombat);

- **Crida:** *L* = DBc. findResultCombatsByCompetitionAndId (cJB, idCombat);
- **Precondició:** -
- **Postcondició:**

Si la competició *compjb* no existeix es llença l'excepció *CompetitionNotExistsException* i *BD=BD'*

Sino *L* és igual als resultats dels combats amb *idCombat* i amb el mateix id de *cJB*. A demás *BD=BD'*.

public **ArrayList<WinnersJB>**
findWinnersByCompetitionAndId(CompetitionJB cJB, int idCombat);

- **Crida:** $L = DBc.findWinnersByCompetitionAndId(cJB, idCombat);$
- **Precondició:** -
- **Postcondició:**
 Si la competició *compjb* no existeix es llença l'excepció *CompetitionNotExistsException* i $BD=BD'$
 Sino L és igual als guanyadors dels combats amb identificador *idCombat* de la competició *cJB* i $BD=BD'$.

private void deleteCombats (CompetitionJB cJB);

- **Crida:** $DBc.deleteCombats(cJB);$
- **Precondició:** -
- **Postcondició:**
 Si la competició no existeix a la BD es llença l'excepció *CompetitionNotExistsException* i $BD=BD'$
 Sino $BD=BD'$ – els combats de la competició *cJB*.

C.6. Operacions De La Classe DBParticipants

public void calculateVictories (CompetitionJB cJB, JudokaJB j1);

- **Crida:** $DBp.calculateVictories(cJB, j1);$
- **Precondició:** -
- **Postcondició:**
 Es calculen i emmagatzement les victories aconseguides pel judoka *j1* en la competició *cJB*.

C.7. Operacions De La Classe DBWeightsCategories

public void addWeightsCategories (WeightsCategories wcJB);

- **Crida:** $DBw.addWeightsCategories(wcJB);$
- **Precondició:** -
- **Postcondició:**
 Si la categoria de pes *wcJB* ja existeix a la BD es llença l'excepció *WeightsCategoriesExistsException* i $BD=BD'$
 Sino $BD=BD'$ Unió *wcJB* (emmagatzema la categoria de pes *wcJB* a la BD).

public void modifyWeightsCategories (WeightsCategoriesJB wcJB);

- **Crida:** DBw.modifyWeightsCategories(wcJB);
- **Precondició:** -
- **Postcondició:**

Si la categoria de pes *wcJB* no existeix a la BD es llença l'excepció *WeightsCategoriesNotExistsException* i *BD=BD'*

Sino la categoria de pes *wcJB* de la BD és modificada amb els nous valors de *wcJB*.

public WeightsCategoriesJB findWeightsCategories (String WeightCategory);

- **Crida:** C = DBw.findWeightsCategories (WeightCategory);
- **Precondició:** -
- **Postcondició:**

Si la categoria de pes identificada per *WeightCategory* no existeix es llença l'excepció *WeightsCategoriesNotExistsException* i *BD=BD'*

Sino *C* és igual a la categoria de pes identificada per *WeightCategory* i *BD=BD'*.

C.8. Operacions De La Classe DBAgeCategories

public ArrayList<AgeCategoriesJB> getAgeCategories ();

- **Crida:** L=DBage.getAgeCategories();
- **Precondició:** -
- **Postcondició:**

L és una *ArrayList* d'objectes *ageCategoriesJB* amb totes les categories d'edats existents a la base de dades i *BD=BD'*.

C.9. Operacions De La Classe DBWinners

public void calculateCombatsWinners (String idCompetition);

- **Crida:** DBw.calculateCombatsWinners(idComp);
- **Precondició:** -
- **Postcondició:**

Calcula els guanyadors dels combats de la competició identificada per *idComp*, segons els seus resultats, i els emmagatzema en la BD.

Annex

D

D. LLICÈNCIA GPL

Els drets de l'aplicació InfoJudo es troben sota la llicència GPL que passem a descriure a continuació.

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

“This License” refers to version 3 of the GNU General Public License.

“Copyright” also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

“The Program” refers to any copyrightable work licensed under this License. Each licensee is addressed as “you”. “Licensees” and “recipients” may be individuals or organizations.

To “modify” a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a “modified version” of the earlier work or a work “based on” the earlier work.

A “covered work” means either the unmodified Program or a work based on the Program.

To “propagate” a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To “convey” a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays “Appropriate Legal Notices” to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a work.

A “Standard Interface” means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The “System Libraries” of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work’s System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to “keep intact all notices”.
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This

License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an “aggregate” if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

“Additional permissions” are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An “entity transaction” is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A “contributor” is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's “contributor version”.

A contributor's “essential patent claims” are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, “control” includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a “patent license” is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To “grant” such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. “Knowingly relying” means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is “discriminatory” if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author>

This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.

This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an “about box”.

You should also get your employer (if you work as a programmer) or school, if any, to sign a “copyright disclaimer” for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <<http://www.gnu.org/licenses/>>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <<http://www.gnu.org/philosophy/why-not-lgpl.html>>.

LLISTAT D'ACRÒNIMS

CSS Cascading Style Sheets
GPL General Public License
HTML Hyper Text Markup Language
HTTP Hypertext Transfer Protocol
ISO International Organization for Standardization
JRE Java Runtime Enviroment
JSP JavaServer Pages
LGPL Lesser General Public License
PHP Hypertext Preprocessor
URL Uniform Resource Locator
UTF Unicode Transformation Format
XML eXtensible Markup Language
XHTML eXtensible Hypertext Markup Language
W3C World Wide Web Consortium

BIBLIOGRAFIA

- [1] Apache Struts: *Struts*.
<http://struts.apache.org/>
- [2] Java Sun: *Java Technology*.
<http://java.sun.com/j2se/1.5.0/docs/api/>
- [3] Java Sun: *Api Java 2 Platform Standard Edition 5.0*.
<http://java.sun.com>
- [4] Java Sun: *Learning the Java Language (The Java Tutorials)*.
<http://java.sun.com/docs/books/tutorial/java/index.html>
- [5] Fundación Sidar - Accesibilidad Universal: Hojas de Estilo en Cascada
<http://www.sidar.org/recur/desdi/mcss/manual/indice.php>
- [6] Galbraith B., Falknet J., Irani R.: *Desarrollo Web con JSP*.
Ed. Anaya Multimedia. 2002
- [7] Wiki – Viquipèdia: *Portada – Viquipèdia*.
<http://ca.wikipedia.org/wiki/Portada>